BAB 2
TINJAUAN PUSTAKA

2.1. Profil Tempat Kerja Praktek

2.1.1. Sejarah Instansi

Terletak di Jl. Selabintana No.37, Sukabumi Jawa Barat, Balai Besar Pengembangan Budidaya Air Tawar (BBPBAT) Sukabumi memiliki sejarah yang panjang sejak jaman penjajahan Belanda. Bermula dari sebuah Sekolah Pertanian yang berdiri pada tahun 1920 yaitu *Culture School/ Landbow School*, dan pada Jaman Jepang lembaga ini sempat berganti nama menjadi *NouGakko*, yang juga berarti sekolah pertanian. Memasuki masa kemerdekaan, dengan peran dan fungsi yang sama, nama lembaga ini menjadi Sekolah Pertanian Menengah.

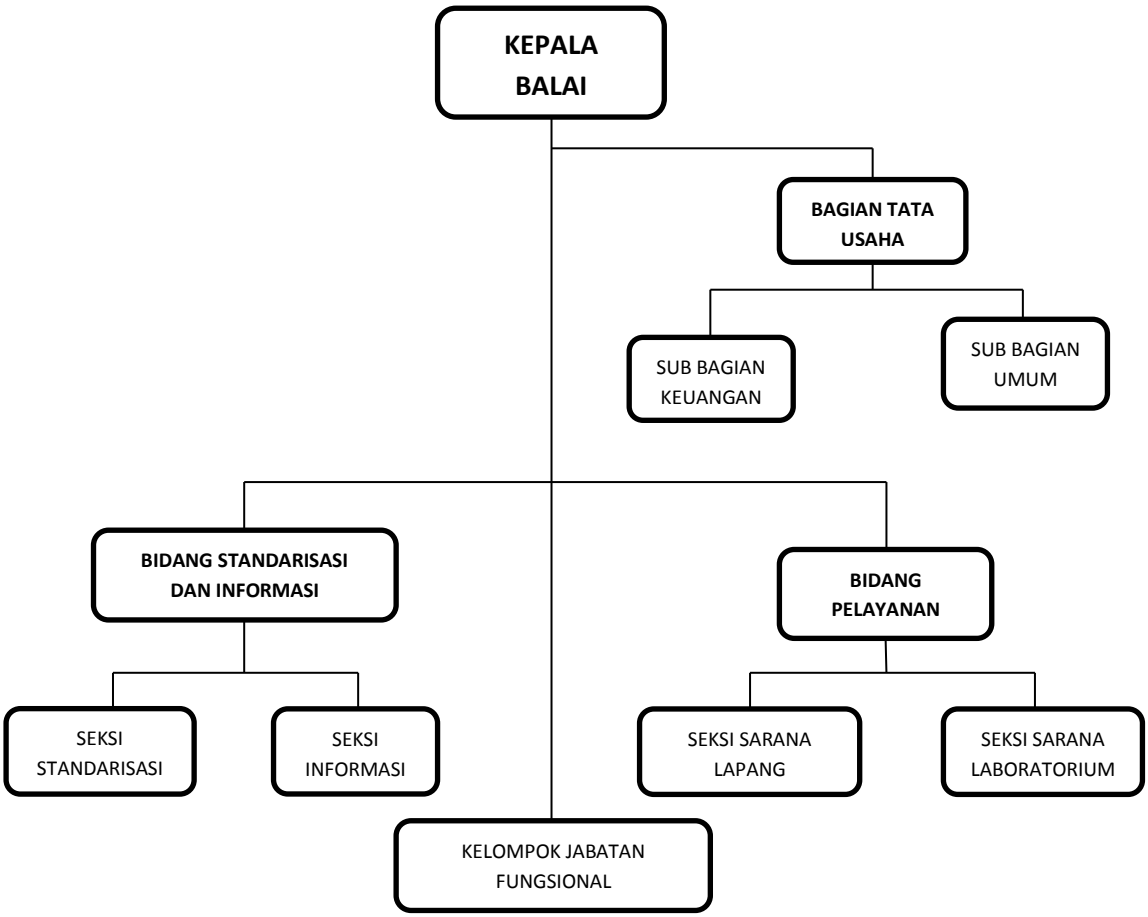
Kiprah dalam dunia perikanan secara khusus dimulai pada tahun 1954 ketika lembaga ini ditetapkan sebagai Pusat Latihan Perikanan. Tahun 1968 sampai dengan tahun 1975 sebutan TC Perikanan (*Training Center* Perikanan) digunakan. Kemudian lembaga ini difungsikan sebagai pangkalan Pengembangan Pola Keterampilan Budidaya Ikan Air Tawar sejak tahun 1976. Peran P3KBAT ditingkatkan ketika pada tahun 1978 secara resmi menjadi Balai Budidaya Air Tawar (BBAT), salah satu pelaksana teknis Direktorat Jenderal Perikanan Departemen Pertanian.

Untuk meningkatkan peran dan fungsi dalam pelaksanaan tugas serta beban kerja yang juga semakin meningkat, pada tanggal 12 Januari 2006 Menteri Kelautan dan Perikanan Nomor Per.06/MEN/2006 yang menetapkan lembaga ini menjadi Balai Besar Pengembangan Budidaya Air Tawar (BBPBAT). Berdasarkan peraturan menteri tersebut, kedudukan BBPBAT adalah sebagai unit pelaksanaan teknis di bidang pengembangan budidaya air tawar yang berada dibawah dan bertanggung

jawab kepada Direktur Jenderal Perikanan, Kementerian Kelautan dan Perikanan.

2.1.2. Struktur Organisasi

Berdasarkan peraturan Menteri Kelautan dan Perikanan Nomor : 06/MEN/2006 Struktur Organisasi BBPBAT Sukabumi adalah sebagai berikut:



Gambar 2.1 Struktur Organisasi BBPBAT Sukabumi

Struktur BBPBAT dilengkapi 160 orang pegawai. Terdiri dari 133 orang PNS serta 27 tenaga kontrak.

2.1.3. Deskripsi Tugas

Selain harus adanya struktur organisasi yang jelas, hal lain yang mendukung mekanisme atau sistem kerja yang baik adalah dengan adanya uraian tugas yang jelas. Berikut ini adalah penjelasan secara garis besar mengenai *job description* dari bagian-bagian yang terkait di BBPBAT Sukabumi :

1. *Kepala*: memimpin penyelenggaraan organisasi dan pembangunan di Balai.
2. *Bagian Tata Usaha*: Mengkoordinasikan penyusunan rencana kerja anggaran dan dokumen pelaksanaan anggaran kantor dan Melaksanakan urusan administrasi kepegawaian antara lain kenaikan pangkat, kenaikan gaji berkala, penggajian pegawai, mutasi dan lain-lain di lingkungan kantor.
3. *Sub Bagian Umum*: membantu Bagian Tata Usaha untuk melaksanakan dan mengkoordinasikan penyiapan bahan perumusan kebijakan, pelayanan teknis administrasi, pengelolaan urusan umum, rumah tangga, pengelolaan barang, kearsipan, pengorganisasian, hukum, dan hubungan kemasyarakatan.
4. *Sub Bagian Keuangan*: mengumpulkan, mengolah dan merumuskan bahan kebijakan penyusunan rencana kerja, rencana strategis, dan pengelolaan tata usaha keuangan serta penyusunan laporan.
5. *Seksi Standarisasi*: penyusunan konsep saran dan pertimbangan kepada Kepala Bidang berkenaan dengan pelaksanaan tugas dan fungsi bidang standarisas.
6. *Seksi Informasi*: Menyusun program pengembangan data dan informasi, Menyusun penyajian data dan informasi dan Melaksanakan tugas-tugas lain yang diberikan oleh Kepala Pemberdayaan Sumberdaya sesuai dengan tugas dan fungsinya.
7. *Seksi Sarana lapang*: menjalankan fungsi pengelolaan dan pelayanan sarana lapangan

8. *Seksi Sarana laboratorium*: menjalankan fungsi pengelolaan dan pelayanan sarana laboratorium.

2.1.4. Logo Instansi



Gambar 2.2 Logo BBPBAT Sukabumi

Maksud dari Logo tersebut adalah BBPBAT Sukabumi bergerak di Bidang perikanan sektor kelautan dan berada di bawah kendali Direktur Jenderal Perikanan Budidaya, Kementerian Kelautan dan perikanan .

2.1.5. Visi

Visi dari BBPBAT Sukabumi ini adalah "Mewujudkan Balai sebagai Institusi Pelayanan Prima dalam Pembangunan dan Pengembangan Sistem Usaha Budidaya Air Tawar yang Berdaya Saing, Berkelanjutan dan Berkeadilan."

2.1.6. Misi

Misi BBPBAT Sukabumi adalah:

1. Meningkatkan kapasitas kelembagan.
2. Mengembangkan rekayasa teknologi budidaya berbasis akuabisnis

dan melaksanakan alih teknologi kepada dunia usaha.

3. Mengembangkan sistem informasi IPTEK perikanan.
4. Meningkatkan jasa pelayanan dan sertifikasi.
5. Memfasilitasi upaya pelestarian sumber daya ikan dan lingkungan.

2.2. Landasan Teori

2.2.1. Data

Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata dan diperoleh dalam bentuk simbol-simbol karakter huruf, angka, gambar, suara, sinyal, dan lain sebagainya.

2.2.1.1.Jenis Data

Adapun jenis data dapat dibedakan menjadi beberapa kategori

1. Berdasarkan sumber data
   a. Data primer

      Data primer adalah secara langsung diambil dari obyek penelitian oleh peneliti perorangan maupun organisasi

   b. Data sekunder

      Data sekunder adalah data yang didapat tidak secara langsung dari objek penelitian.
2. Berdasarkan tampilan data
   a. Data Kuantitatif

      Data kuantitatif adalah data yang dipaparkan dalam bentuk angka-angka.

   b. Data Kualitatif

      Data kualitatif adalah data yang disajikan dalam bentuk kata-kata yang mengandung makna.

2.2.2. Sistem

Sistem merupakan kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan dalam usaha mencapai satu tujuan[4]. Di dalam perusahaan, yang dimaksud sistem adalah

departemen internal, seperti persediaan barang mentah, produksi, persediaan barang jadi, promosi, penjualan, keuangan, personalia, serta pihak internal dan eksternal seperti *supplier* dan konsumen yang saling terkait satu sama lain dan membentuk kesatuan usaha.

2.2.3. Informasi

Informasi merupakan data yang telah diklasifikasikan atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan Informasi dapat juga diartikan data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi[4]. Data merupakan nilai, keadaan, atau sifat yang berdiri sendiri lepas dari konteks apapun. Sementara informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang[4]. *Mc Leod* mengatakan bahwa informasi adalah data yang telah diproses, atau data yang memiliki arti[4].

2.2.4. Pengertian Sistem Informasi

Menurut *Robert A. Laitch* dan *K. Roscoe Bavis* Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat managerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan [4].

2.2.4.1.Fungsi Sistem Informasi

Adapun beberapa kemampuan sistem informasi adalah sebagai berikut:

1. Komputasi numerik, memiliki volume besar, dengan kecepatan tinggi
2. Menyediakan komunikasi dalam organisasi atau antar organisasi yang murah akurat dan cepat.

3. Menyimpan informasi dalam jumlah yang sangat besar dalam ruang yang kecil tetapi mudah diakses.
4. Memungkinkan pengaksesan informasi yang sangat banyak di seluruh dunia dengan cepat dan murah.

2.2.5. *Microsoft Visual Basic*

Microsoft Visual Basic (VB) merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment* (IDE) *visual* untuk membuat program perangkat lunak berbasis sistem operasi Microsoft Windows dengan menggunakan model pemrograman (COM), *Visual Basic* merupakan turunan bahasa pemrograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat, Beberapa bahasa skrip seperti *Visual Basic for Applications* (VBA) dan *Visual Basic Scripting Edition* (VBScript), mirip seperti halnya *Visual Basic*, tetapi cara kerjanya yang berbeda.

Para programmer dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh *Microsoft Visual Basic* Program-program yang ditulis dengan *Visual Basic* juga dapat menggunakan *Windows API*, tapi membutuhkan deklarasi fungsi luar tambahan. Dalam pemrograman untuk bisnis, *Visual Basic* memiliki pangsa pasar yang sangat luas.

2.2.6. Pengertian Microsoft Access

Microsoft Access adalah sistem menggunakan *database* relasional yang bisa mengatur data tentang subjek-subjek yang berbeda kedalam tabel-tabel kemudian bisa membuat hubungan diantara tabel. Dalam perancangan perangkat lunak pengujian *Memory span* (rentang ingatan) yang akan dirancang adalah database yang dapat mempermudah didalam pembuatan program. Dimana keseluruhan *database* ini dirancang dengan menggunakan *Microsoft Access*.

Dalam *Microsoft Access* terdapat berbagai objek antara lain:

1. Tabel

   objek *database* yang digunakan untuk menyimpan data-data tentang subjek tertentu.

2. *Query*

   objek *database* yang digunakan untuk meminta informasi kepada tabel dan menampilkan informasi yang diminta.

3. *Form*

   objek *database* yang digunakan untuk menampilkan data dalam tabel dengan bentuk lain atau formulir.

4. *Report*

   objek *database* laporan yang dibuat dengan menggunakan objek tabel, database (*query*).

5. *Macro*

   objek *database* berupa aksi yang akan didefinisikan sebelumnya.

6. *Module*

   objek *database* berupa aksi yang didefinisikan sebelumnya karena lebih kompleks.

2.2.7. *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan

mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem[1].

UML terdiri atas banyak elemen-elemen grafis yang digabungkan membentuk diagram. Tujuan representasi elemen-elemen grafis ke dalam diagram adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut. Kumpulan dari beragam sudut pandang inilah yang disebut sebuah model.

2.2.7.1. *Use Case Diagram*

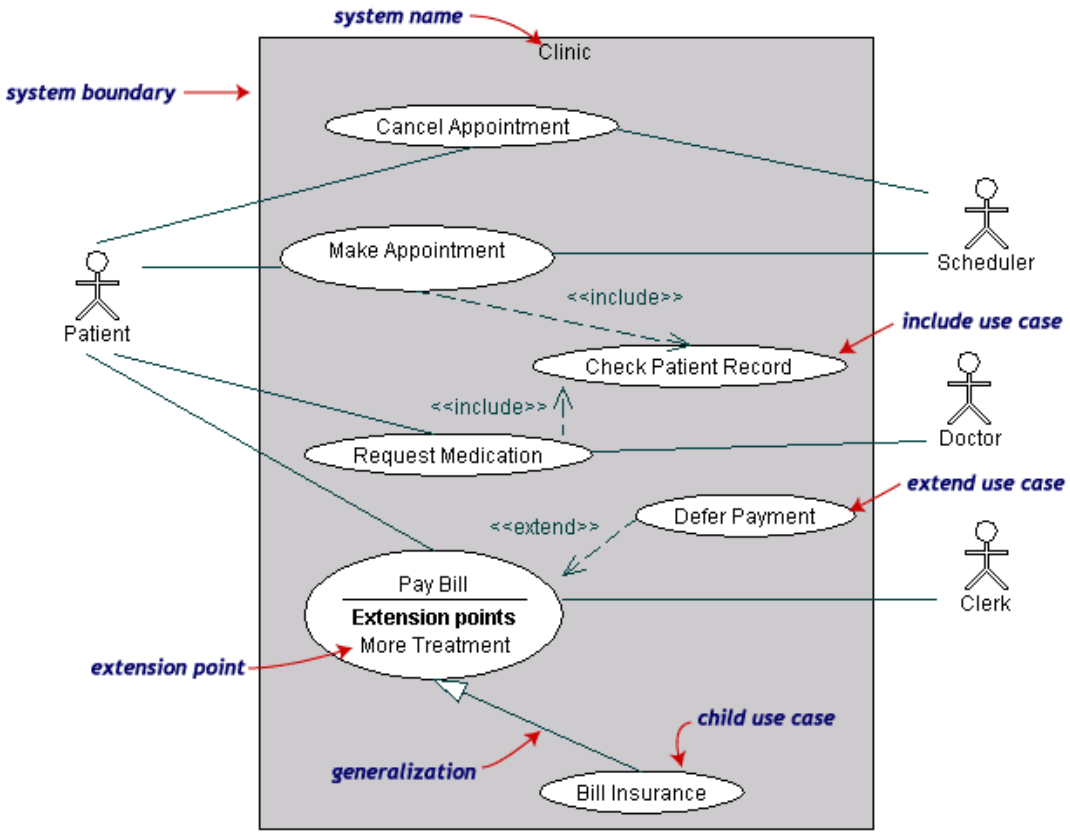
Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah apa yang diperbuat sistem, dan bukan bagaimana. Sebuah *use case* merepresentasikan sebuah interaksi antara *actor* dengan sistem[1]. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, menmbuat sebuah daftar belanja, dan sebagainya.

Seorang/sebuah *actor* adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua fitur yang ada pada sistem.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behavior*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Contoh *use case diagram* :



Gambar 2.3 Contoh *use case diagram*

2.2.7.2. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi)[1].

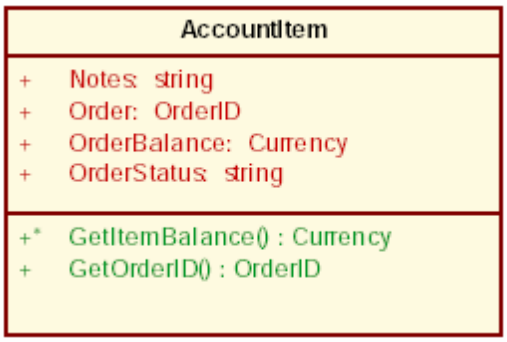
Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain[1].

Class memiliki tiga area pokok :

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja.



Gambar 2.4 Atribut dan Metoda dalam *Class Diagram*

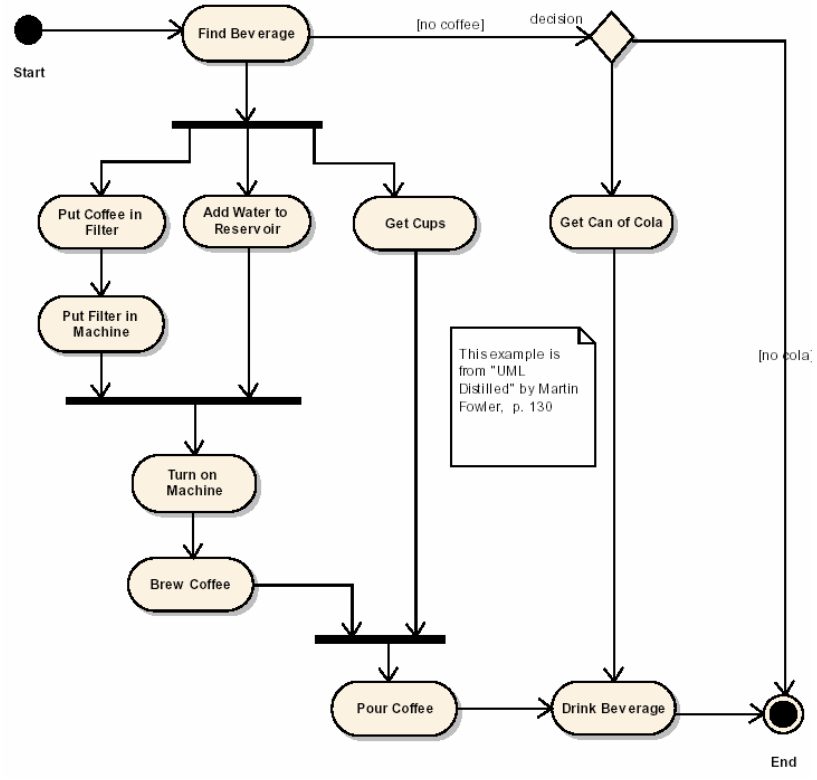
Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.

2.2.7.3. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi[1].

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan prosesproses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.



Gambar 2.5 Contoh *Activity Diagram* Tanpa Swimlane

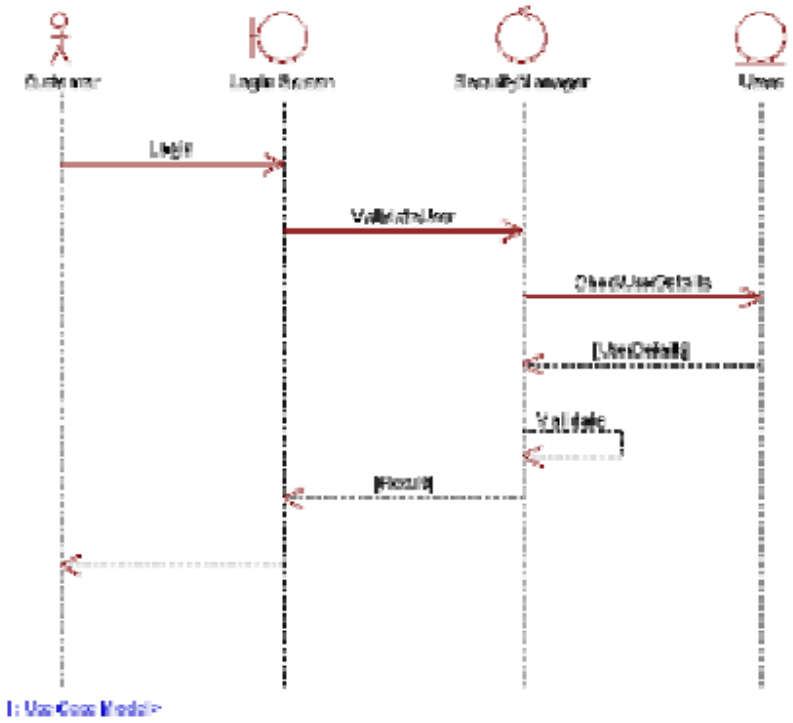
2.2.7.4. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait)[1].

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*.

Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.

Contoh *sequence diagram* :



Gambar 2.6 Contoh *Sequence Diagram*